NeXus File Specification
Issues

May 2, 2002
Gary Cooper, Tom Kozlowski, Thomas Proffen
Los Alamos Neutron Scattering Center

TABLE OF CONTENTS

1 General

1.1 Issues

1.1.1 NeXus standard organization

It would be useful to have a single document that contains the official declaration of the NeXus standard. This document should explicitly state the version (and release date?) of the standard. The contents of the document should be written in a fairly "legalistic" manner, such that it is clear what is mandatory, what is recommended (but not required), and what is left up to the user/implementer. A bulleted or numbered list format can be helpful. A useful convention in standard declarations is the use of the word "*shall* " for mandatory usage and "*should*" for recommended usage. See the NXdata section for an example.

1.1.2 Meta-DTD format element descriptions

In addition to the meta-DTD descriptions of class structures and a short paragraph describing the class purpose, a short descriptive paragraph should be provided for a*ll* class elements. The meta-DTD descriptions of class elements are too short to clearly and unambiguously define them. In addition any rules of usage of elements and relations between them should be clearly stated. See the NXdata section for an example.

1.1.3 Clear statements of mandatory usage

The meta-DTD descriptions are often not sufficient for defining the contents and constraints on elements of a group class. For example there are constraints on the dimensions of the "dimension arrays" in an NXdata group. The group class definitions should explicitly state any such constraints or restrictions (perhaps in a bulleted list) and whether they are mandatory or not. See the NXdata section for an example.

1.1.4 Clear statements of recommended usage

Clear statements of recommended usage even if not required (perhaps in a bulleted list) could be useful to developers of NeXus applications. See the NXdata section for an example.

1.1.5 Meta-DTD format optional attributes

The format description does not seem to have a notation for indicating optional attributes. Is the implication that they are required? Maybe the notation of a "?" after the definition of an attribute could be used to indicate an optional attribute.

1.1.6 Meta-DTD format attribute values

The format description implies that a literal string (without curly brackets) after the "=" in an attribute definition mandates that the value of the attribute *must be* the literal string. If this is the case, it should be clearly stated.

1.1.7 Meta-DTD format group and data element names

While the format description explicitly states that every NeXus group has a "name" attribute that maps to the identification label attached to a Vgroup when it is created, nothing is said about the names of data elements (scientific data sets). For the sake of consistency, shouldn't there also be a name attribute for a data set which maps to the "data_name" label of the data set? It is true that in most cases within a group class definition the names of data elements are unambiguous and can be the name of the element used in the DTD. The format description hints at this, but it should state this rule explicitly. There are cases however, where there can be more than one data element of a particular type ("variable" in the NXdata class for example) where the names will have to be different. Use of a name attribute in the DTDs would help in making this clearer. See the NXdata section for an example.

1.1.8 When are HDF links permitted?

It should be clearly stated that in a NeXus group, a group element (NeXus group or data set) can be replaced by a link at any time? This is the case – correct?

1.1.9 Recommend a standard default coordinate system

The NeXus standard should define a standard coordinate system that is to be used as the default for geometrical elements. A proposal is (1) a spherical coordinate system, (2) oriented with the z (polar) axis in the beam direction, y up, and the origin at the nominal sample position. The coordinates would then be: radius, theta (or polar_angle) and phi (or azimuthal_angle). For non-default coordinate systems, maybe an attribute(s) could be added at the file level that contains the name of the coordinates system used in the file.

1.1.10 Recommend standard default units

The NeXus standard should define standard units for most physical elements to be used as the default when the units attribute is not explicitly defined. Proposal: centimeters, grams, microseconds (nanoseconds?), radians? Any deviations should make use of the units attribute.

2 NeXus File Structure

```
<NeXus_file
        file_name=”{File name of original NeXus file}”
        file_time=”{ISO8601 Creation time of file}”
        NeXus_version=”{Program version number}”
        user=”{Name of user responsible for producing the file}”
        affiliation=”{User's affiliation}”
        address=”{User's postal address (complete)}”
        telephone_number=”{User's telephone number}”
        fax_number=”{User's fax number}”
        email=”{User's e-mail address}”
      >
      <NXentry name="{Entry Name}">+</NXentry>
  </NeXus_file>
```

2.1 Issues

2.1.1 Multiple users

It would be useful to allow multiple users, both at the file level and at the NXentry level. The user related global attributes as currently defined do not map well to this need. A suggestion is allow multiple NXuser groups at the file (root) level. That might be useful anyway even if multiple users where not allowed. An attribute could be added to NXuser to indicated a “primary user”. If this was the case the user related file level attributes could be dispensed with.

2.1.2 File level mandatory attributes

It is not clear if any of the file level attributes are mandatory.

2.1.3 No restrictions on groups and data sets at file level

The standard does not state explicitly whether or not any NeXus groups or data sets other than NXentry instances may be present at the file (root) level. In some scenarios, it could be useful to put common groups or data sets at the root level, which are in turn linked to/from multiple NXentry groups. For example, in a file containing several NXentry in which the instrument descriptions don’t change, each NXentry instance may refer to the same NXinstrument and its component instances.

3 NXentry

```
<NXentry name="{Entry Name}">
   <title>{Extended title for entry}</title>
   <analysis version="{DTD version number}" URL="{URL of DTD file}">{Name of entry
       DTD}</analysis>
   <start_time type="ISO8601">{Starting time of measurement}?</start_time>
   <end_time type="ISO8601">{Ending time of measurement}?</end_time>
   <duration type="NX_INT32" units="seconds">{Duration of measurement}?</duration>
   <run_number type="NX_INT32">{Number of run or scan stored in this
       entry}?</run_number>
   <program_name version="{Program version number}">{Name of program used to
       generate this file}?</program_name>
   <command_line>{Name of command line used to generate this file}?</command_line>
   <notes>{Notes describing entry}</notes>
   <NXuser name="user">?</NXuser>
   <NXsample name="sample">?</NXsample>
   <NXinstrument name="{Name of instrument}">?</NXinstrument>
   <NXmonitor name="{Name of monitor}">*</NXmonitor>
   <NXdata name="{Name of data block}">+</NXdata>
</NXentry>
```

3.1 Issues

3.1.1 "Analysis" attribute name misleading

Would "template" or "document_type" or "DTD" be a better name for this attribute?

3.1.2 Recommend a standard NXentry name

Recommend that the title attribute should be in the form "run_1234" where "1234" is the run number for that entry.

3.1.3 Multiple users

It would be useful to allow multiple users, both at the file level and at the NXentry level, since many experiments have several investigators. An attribute could be added to NXuser to indicate "primary " user(s) or investigator(s). If multiple users are present, then the name attribute should be set equal to the user's last name (or "user_" suffixed by the last name?).

3.1.4 Add author and date to notes

Add "author" and "date" information to a note . This would allow notes to be used much like "logbook" entries. This could be done by adding "author" and "date" attributes to a note data set, or by adding "author" and "date" data elements to a new NXnote class. The latter is preferred if notes are generalized to include graphical objects (next item):

```
<NXnote name="{Identifier: author_date_n?}">
    <author>{Authors name or account name}</author>
    <date type="ISO8601">{Data and time note originally  created}</date>
    <type>"textual"</type>
    <note >{Note contents}</note>
 </NXdata>
```

3.1.5 Generalize notes to be graphical as well as textual

In the spirit of using notes like logbook entries, it would be useful to attach graphs and pictures as notes.

3.1.6 Allow multiple notes to be attached to any NXgroup or file

By their very nature it would be useful to be able to attach notes to any NeXus class, not just NXentry classes.

4 NXdata

```
<NXdata name="{Name of data block}">
    <variable name=”{Axis name}”  type="NX_FLOAT32[:]|NX_INT32[:]"
        long_name="{Axis label}">{Dimension scale defining an axis of the data}+</variable>
    <data name=”data”, type="NX_FLOAT32[:...]|NX_INT32[:...]" signal="1" axes="[...]"
        long_name="{Title of data}">{Data values}</data>
    <errors name=”errors”, type="NX_FLOAT32[:...]">{Standard deviations of data values.
        This must have the same dimensions as the data}?</errors>
 </NXdata>
```

variable
“Dimension scales” for an independent variable axis for a possibly multi-dimensional histogram or data array.

- There ***shall*** be at least one “variable” data set for each independent variable (axis) for the data.
- The dimension of the “variable” array ***shall*** be either the same as or one more than the dimension of the data (signal) array.
- The “axis” attribute of a “variable” data set ***shall*** be set to 1, if this “variable” data set corresponds to the fastest changing index for (signal) data represented as a multi-dimensional array; it ***shall*** be set to 2, it it corresponds to the second fastest changing index; etc.
- If the (signal) data *is not* histogram data, “histogram_offset” ***shall*** be set to zero. In this case the dimension of the array ***shall*** be the same as the dimension of the signal array.
- If the (signal) data *is* histogram data, and if the “variable” array for this axis contains the values for histogram bin centers, the dimension ***shall*** be the same as the dimension of the data (signal) array. The “histogram_offset” ***shall*** be set to the distance between the lowest histogram bin boundary and the center of the first histogram bin.
- If the (signal) data *is* histogram data and if the “variable” array for this axis contains the values for histogram bin boundaries, the dimension ***shall*** be one more than the dimension of the data (signal) array. The “histogram_offset” ***shall*** be ignored.
- It is recommended that the name of the group be the same as “long_name”.

data
The actual possibly multi-dimensional histogram or data array.

- There ***shall*** be one and only one “data” data set for each NXdata group instance.
- The number of array indices ***shall*** be the same as the number of axes (variable data sets with unique axis attributes), with the fastest changing index corresponding to axis=1, so on.

errors
The standard deviations for the data array.

- The dimension of the errors array ***shall*** be the same as the data array.

4.1 Issues

4.1.1 New definitions for “data” and “variable” data sets?

The original NeXus file definitions defined “axis”, “histogram_offset and “primary” attributes for the “variable” data set, and “calibration_status” for the “data” data set. These are now gone. Has the NXdata group been completely redefined? What is “axes”? More details are needed. In the following we assume the original attributes are still there. This is an example while the standard should always explicitly describe all attributes and whether they are optional or not.

4.1.2 *Need explicit statement of usage rules:*

In the spirit of a more formal standards document, text like that which appears above in the data element definitions after the NXdata meta-DTD should be added, since the DTD is not sufficient to define these rules.

4.1.3 *The standard should allow either "bin boundary" or "bin center" variable arrays:*

Rules that would implement this appear in the data element definitions after the NXdata meta-DTD.

4.1.4 *Category attribute:*

Some instrument data acquisition systems generate many different histograms for various purposes, including diagnostics. For example, these might include: time-of-flight for 10 degree bank, time-of-flight for 20 degree bank, pulse-height v.s. pulse-height v.s. detector number (PSDs), longitudinal-position v.s. detector number, monitor pulse-height. In order for analysis tools to identify histograms of interest for a particular purpose, an NXdata "category" attribute would be useful. For example it might be set to "time_of_flight" or "pulse_height", "monitor_pulse_height", etc.

4.1.5 *Array notation*

There is no definition of the "[:…]" array notation. From the context it looks like it is meant to indicate multidimensional arrays of indeterminate rank and dimension. Maybe this type of information is best described in the "rules of usage".

4.1.6 *Data type*

The previous standard definitions defined the time-of-flight and dimension data arrays to be of type NX_FLOAT32. The current description allows more flexibility. It is OK with us, but it does complicate generic analyis software that must read the NeXus files. Maybe NX_FLOAT32(?) needs to be the recommended type.

5 NXlog

```
<NXlog name="{Name of logged measurements}">
    <time type="ISO8601[i]">{Time of logged entry}</time>
    <temperature type="NX_FLOAT32[i]" units="K">{Sample temperature}?</temperature>
    <electric_field type="NX_FLOAT32[i]" units="V">{Applied electric
        field}?</electric_field>
    <magnetic_field type="NX_FLOAT32[i]" units="T">{Applied magnetic
        field}?</magnetic_field>
 </NXlog>
```

5.1 Issues

5.1.1 Any variables should be allowed in the log

Temperature, electric_field and magnetic field are too specific. A single log file should be able to contain arbitrary monitored variables. We suggest a definition more like:

```
<NXlog name="{Name of logged measurements}">
    <time type="ISO8601[i]">{Time of logged entry}</time>
    <logged_variable name=”{Name of variable}”  type=”NX_FLOAT32[i]" units="{}"
        >{Monitored variable data}+</ logged_variable >
 </NXlog>
```

6 NXuser

```
<NXuser name="user">
    <name>{Name of user responsible for this entry}</name>
    <affiliation>{Affiliation of user}?</affiliation>
    <address>{Address of user}?</address>
    <telephone_number>{Telephone number of user}?</telephone_number>
    <fax_number>{Fax number of user}?</fax_number>
    <email>{Email of user}?</email>
 </NXuser>
```

6.1 Issues

6.1.1 Multiple users

If multiple users in an NXentry (or NeXus file) are allowed, the *name* attribute should be the name of the user (maybe the last name or “user_” followed by the last name). In addition a “primary” attribute, if present, could be used to flag principle users or investigators.

7 NXsample

```
<NXsample name="sample">
    ?
    <name>{Descriptive name of sample}</name>
    <chemical_formula>{Chemical formula}?</chemical_formula>
    <temperature type="NX_FLOAT32[:]">{Sample temperature. This could be a scanned
         variable}?</temperature>
    <electric_field type="NX_FLOAT32[3]">{Applied electric field}?</electric_field>
    <magnetic_field type="NX_FLOAT32[3]">{Applied magnetic field}?</magnetic_field>
    <stress_field type="NX_FLOAT32[3]">{External stress}?</stress_field>
    <pressure type="NX_FLOAT32[:]">{Applied pressure}?</pressure>
    <changer_position type="NX_INT32">{Sample changer position}?</changer_position>
    <unit_cell type="NX_FLOAT32[6])">{Unit cell parameters (lengths and
         angles)}?</unit_cell>
    <unit_cell_volume type="NX_FLOAT32" units="Angstroms3" rank="1">{Volume of the
         unit cell}?</unit_cell_volume>
    <orientation_matrix type="NX_FLOAT32[9]">{Orientation matrix of single crystal
         sample}?</orientation_matrix>
   <symmetry_cell_setting>"cubic"|"tetragonal"|"orthorhombic"|"monoclinic"|"triclinic"?</sym
         metry_cell_setting>
    <shape>"plate"|"sphere"|"cylinder"|"hollow cylinder"?</shape>
    <dimension type="NX_FLOAT32[3]" units="cm">{Dimensions of sample if plate-
         like}?</dimension>
    <radius type="NX_FLOAT32" units="cm">{Radius of sample if cylindrical or
         spherical}?</radius>
    <inner_radius type="NX_FLOAT32" units="cm">{Inner radius of sample if
         hollow}?</inner_radius>
    <height type="NX_FLOAT32" units="cm">{Height of sample}?</height>
    <mass type="NX_FLOAT32" units="g">{Mass of sample}?</mass>
    <density type="NX_FLOAT32" units="g cm-3">{Density of sample}?</density>
    <molecular_weight type="NX_FLOAT32">{Molecular weight of
         sample}?</molecular_weight>
    <coherent_cross_section type="NX_FLOAT32" units="barns">{Coherent cross
         section}?</coherent_cross_section>
    <incoherent_cross_section type="NX_FLOAT32" units="barns">{Incoherent cross
         section}?</incoherent_cross_section>
    <absorption_cross_section type="NX_FLOAT32" units="barns">{Absorption cross
         section}?</absorption_cross_section>
    <NXlog name="{Name of logged measurements}">*</NXlog>
 </NXsample>
```

7.1 Issues

7.1.1 Superfluous "?"

What is the meaning of the "?" following "<NXsample name="sample">."

7.1.2 The sample contents should be "user" defined

Real world sample and sample environments tend to be very specific to a specific instrument and or a specific experiment. It is almost impossible to specify a standard description that will satisfy everyone's needs. Perhaps a very generic definition in the standard would be better. For example:

```
<NXsample name="sample">
    <name>{Descriptive name of sample}</name>
    <sample_parameter  name=”{Name of parameter}”  type="{Data type}"  units="{}"
        dimension=”{}” >{Parameter that describes the sample}*</sample>
    <NXlog name="{Name of logged measurements}">*</NXlog>
 </NXsample>
```

Alternatively an NXentry instance could reference a user, instrument or experiment specific sample description meta-DTD.

8 NXmonitor

```
<NXmonitor name="{Name of monitor}">
    <distance type="NX_FLOAT32" units="m">{Distance of monitor from sample}</distance>
    <integral type="NX_FLOAT32">{Integral over monitor spectrum}?</integral>
    <range type="NX_FLOAT32[2]">{Time-of-flight range over which the integral was
        calculated}?</range>
    <type>"Fission Chamber"|"Scintillator"?</type>
    <height type="NX_FLOAT32" units="cm">{Height of monitor}?</height>
    <width type="NX_FLOAT32" units="cm">{Width of monitor}?</width>
    <time_of_flight type="NX_FLOAT32[i]" units="microseconds">{Time-of-
        flight}?</time_of_flight>
    <efficiency type="NX_FLOAT32[i]">{Monitor efficiency}?</efficiency>
    <data type="NX_INT32[i]">{Monitor data}?</data>
 </NXmonitor>
```

8.1 Issues

8.1.1 Monitors are just a specific type of detector

Monitor descriptions tend to be very similar if not identical to normal detector descriptions. What about multiple element monitors? We could imagine dispensing with the separate monitor class altogether, using the detector class for monitors. They could be distinguished by adding an “is_monitor” attribute to NXdetector (preferred) or by the name of the detector group.

8.1.2 Monitor location

Rather than just *distance,* the location should be given in 3-dimensional space (recommend: spherical coordinates of radius, polar angle and aximuthal angle). It should be similar to the NXdetector class, since a monitor is just a specific type of or use of a detector.

8.1.3 Monitor dimensions

This definition explicitly uses height and width, what about circular (or elliptical) monitors. A rule that circular monitors shall use the height element to store the radius, might suffice.

8.1.4 Mandated units?

The description seems to mandate (absence of curly brackets) specific units of “m”, “cm”, and “microseconds”. Is this really the intent? Propose either the default units mentioned previously or implementer defined units.

8.1.5 Monitor data

The NXmonitor “time_of_flight” and “data” arrays are used in a similar manner to the NXdata “data” and “variable” arrays. For the sake of consistency, these data sets should have the same attributes and types as the NXdata data sets. To carry this idea further, why not dispense with the “time_of_flight” and “data” arrays and replace them with an NXdata group (or link to an NXdata group already present at the NXentry level)? We also have monitor pulse-height spectra, which we would like to access in a similar manner. The use of the NXdata class for any plottable data arrays may also make display tools simpler (more uniform).

9 NXinstrument

```
<NXinstrument name="{Name of instrument}">
    <NXsource name="{Name of facility}">?</NXsource>
    <NXchopper name="{Name of chopper}">*</NXchopper>
    <NXcrystal name="{Name of crystal monochromator or analyzer}">*</NXcrystal>
    <NXaperture name="{Name of beamline aperture}">*</NXaperture>
    <NXcollimator name="{Name of collimator}">*</NXcollimator>
    <NXattenuator name="{Name of beam attenuator}">*</NXattenuator>
    <NXpolarizer name="{Name of beam polarizer}">*</NXpolarizer>
    <NXflipper name="{Name of beam polarization flipper}">*</NXflipper>
    <NXmirror name="{Name of beam guide mirror}">*</NXmirror>
    <NXdetector name="{Name of detector, bank of detectors, or
        multidetector}">+</NXdetector>
 </NXinstrument>
```

9.1 Issues

9.1.1 NXinstrument class unnecessary?

The NXinstument class seems to be a somewhat arbitrary grouping together associated with a specific name. It is not entirely , for example, why the sample and monitor classes are not included within this instrument class. Why not simply dispense with the instrument class and move its contents to the the NXentry level? Perhaps the instrument name itself could be implemented as a file root level attribute ("instrument").

10 NXsource

```
<NXsource name="source">
    <distance type="NX_FLOAT32" units="m">{Distance from sample}</distance>
    <name>{Name of facility}?</name>
    <type>"Spallation Neutron Source"|"Pulsed Reactor Neutron Source"|"Reactor Neutron
         Source"|"Synchrotron X-ray Source"?</type>
    <power type="NX_FLOAT32" units="MW">{Source power}?</power>
    <proton_current type="NX_FLOAT32" units="microamps">{Accelerator proton
         current}?</proton_current>
    <proton_voltage type="NX_FLOAT32" units="MeV">{Accelerator proton
         voltage}?</proton_voltage>
    <frequency type="NX_FLOAT32" units="Hz">{Frequency of pulsed source}?</frequency>
    <period type="NX_FLOAT32" units="microseconds">{Period of pulsed source}?</period>
    <target_material>{Pulsed source target material}
         "Ta"|"W"|"depleted_U"|"enriched_U"|"Hg"|"Pb"|...?</target_material>
    <moderator>"H2O"|"D2O"|"H2"|"CH4"|...?</moderator>
    <moderator_temperature type="NX_FLOAT32" units="K">{Temperature of
         moderator}?</moderator_temperature>
    <poison_depth type="NX_FLOAT32" units="cm">{Poison depth of
         moderator}?</poison_depth>
 </NXsource>
```

10.1 Issues

10.1.1 "Current" and "voltage"

In order to be generic, leave "proton" off of "current" and "voltage".

10.1.2 Integrated current.

For spallation sources, at least, a data element containing the value of the "integrated_current" for a run is useful. This may be of use for light source facilities also (?).

10.1.3 "Frequency" and "period"

What exactly is meant by these terms? If they refer to the same physical entity, the repetition frequency of the source), only one of them should be defined as the standard (frequency?).

10.1.4 Source pulse width.

For spallation sources, at least, a data element containing the value of the "pulse_width" of the proton pulse on target is useful.

11 NXdetector

```
<NXdetector name="{Name of detector bank}">
    <id type="NX_INT32[i]">{Identifier of detector element}</id>
    <distance type="NX_FLOAT32[i]" units="m">{Distance from sample}</distance>
    <polar_angle type="NX_FLOAT32[i]" units="degrees">{Polar angle of detector
        element}?</polar_angle>
    <azimuthal_angle type="NX_FLOAT32[i]" units="degrees">{Azimuthal angle of detector
        element}?</azimuthal_angle>
    <solid_angle type="NX_FLOAT32[i]" units="steradians">{Solid angle subtended by the
        detector at the sample}?</solid_angle>
    <height type="NX_FLOAT32[i]" units="cm">{Height of detector element}?</height>
    <width type="NX_FLOAT32[i]" units="cm">{Width of detector element}?</width>
    <depth type="NX_FLOAT32[i]" units="cm">{Depth of detector element}?</depth>
    <orientation type="NX_FLOAT32[i,6]">{Orientation of detector}?</orientation>
    <gas_pressure type="NX_FLOAT32[i]" units="bars">{Detector gas
        pressure}?</gas_pressure>
    <crate type="NX_INT32[i]">{Crate number of detector}?</crate>
    <slot type="NX_INT32[i]">{Slot number of detector}?</slot>
    <input type="NX_INT32[i]">?</input>
    <type>"He3 gas cylinder"|He3 PSD"|"He3 multidetector"|"BF3 gas"|"scintillator"|"fission
        chamber"?</type>
    <time_of_flight type="NX_FLOAT32[i]" units="microseconds">{Neutron time-of-
        flight}?</time_of_flight>
    <efficiency type="NX_FLOAT32[i]">{Efficiency of detector}?</efficiency>
 </NXdetector>
```

11.1 Issues

11.1.1 Mandated units?

The description seems to mandate (absence of curly brackets) specific units of "m", "cm", and "microseconds". Is this really the intent? Propose either the default units mentioned previously or implementer defined units.

11.1.2 Orientation

What is the definition of the orientation components? Why are six quanities needed? For example, three euler. angles could suffice (The standard could recommend specific set of euler angles.).

11.1.3 Some detector parameters are constant for a bank of detectors

Except for "type", the description mandates that each detector parameter is an array, each of the same size. For a bank of detector tubes, several data elements are constant for an entire bank. These are typically: height, width, depth, orientation, gas_pressure, type. A scintillator bank would be similar. The detector parameters listed above should be (optionally?) scalar quantities that apply to the entire bank.

11.1.4 Type

The description appears to mandate specific enumerated values. Is this what is really wanted, or would a simple text string defined by the instrument/facility sufficient? For one thing, we may want to add the manufactures name to the type name.

11.1.5 Efficiency "calibration_status"

Efficiency should have a "calibration_status" attribute to indicate that the efficiencies are "nominal" or "measured".

11.1.6 Position "calibration_status"

Distance, polar_angle and azimuthal_angle should have "calibration_status" attributes to indicate that they they are "nominal" or "measured".

11.1.7 What is the time_of_flight element?

This appears to be a time of flight "dimension array", but there is no corresponding data array. If it is intended to refer to the histogram data, for the sake of consistency, these data sets should have the same attributes and types as the NXdata data sets. To carry this idea further, why not dispense with the "time_of_flight" and "data" arrays and replace them with an NXdata group (or link to an NXdata group already present at the NXentry level)? We also have monitor pulse-height spectra, which we would like to access in a similar manner. The use of the NXdata class for any plottable data arrays may also make display tools simpler (more uniform).

11.1.8 Detector related histograms

There may be several detector associated histograms, each with different structures. For example for PSD detectors, the time of flight histogram may be a 3-d histogram (id v.s. longitudinal_position v.s. tof) or a pulse-height histogram (id v.s. pulse-height-1 v.s. pulse-height-2). Because the histograms can vary with different types of detectors, it may make sense, in addition to the reason described in the previous item, to include multiple NXdata group elements.

12 NXchopper

```
<NXchopper name="{Name of chopper}">
    <distance type="NXFLOAT32" units="m">{Distance of chopper from sample}</distance>
    <type>"Fermi"|"disk"|"counter rotating disk"|"velocity selector"?</type>
    <frequency type="NX_FLOAT32" units="Hz">{Frequency of chopper
        rotation}?</frequency>
    <period type="NX_FLOAT32" units="microseconds">{Period of chopper
        rotation}?</period>
    <radius type="NX_FLOAT32" units="cm">{Radius of chopper body}?</radius>
    <curvature type="NX_FLOAT32" units="cm">{Radius of curvature of Fermi
        chopper}?</curvature>
    <slit_width type="NX_FLOAT32" units="cm">{Width of Fermi chopper
        slits}?</slit_width>
    <blade_width type="NX_FLOAT32" units="cm">{Width of Fermi chopper
        blades?</blade_width>
    <slit_number>{Number of Fermi chopper slits}?</slit_number>
    <energy type="NX_FLOAT32" units="meV"
        calibration_status="'Nominal'|'Measured'">{Optimum energy transmitted by the
        chopper}?</energy>
    <NXlog name="trigger_log">{Log of chopper trigger pulses}?</NXlog>
    <NXlog name="phasing_log">{Log of chopper phases}?</NXlog>
 </NXchopper>
```

12.1 Issues

12.1.1 Nominal and measured energies

It may be of use to carry around both the nominal and measured energies. In that case, two instances of the energy data element are required.

13 NXcrystal

```
<NXcrystal name="{Name of crystal beamline component}">
    <distance type="NXFLOAT32" units="m">{Distance of chopper from sample}</distance>
    <wavelength type="NXFLOAT32" units="Angstroms">{Optimum diffracted
        wavelength}?</wavelength>
    <energy type="NXFLOAT32" units="meV">{Optimum diffracted energy}?</energy>
    <lattice_parameter type="NX_FLOAT32" units="Angstrom">{Lattice parameter of the
        nominal reflection}?</lattice_parameter>
    <reflection type="NX_INT32[3]">{[hkl] values of nominal reflection}?</reflection>
    <horizontal_curvature type="NX_FLOAT32" units="degrees">{Horizontal curvature of
        focusing crystal}?</horizontal_curvature>
    <vertical_curvature type="NX_FLOAT32" units="degrees">{Vertical curvature of focusing
        crystal}?</vertical_curvature>
    <horizontal_aperture type="NX_FLOAT32" units="cm">{Horizontal aperture, if
        rectangular}?</horizontal_aperture>
    <vertical_aperture type="NX_FLOAT32" units="cm">{Vertical aperture, if
        rectangular}?</vertical_aperture>
 </NXcrystal>
```

13.1 Issues

14 NXcollimator

```
<NXcollimator name="{Name of collimator}">
    <distance type="NX_FLOAT32" units="m">{Distance from sample}</distance>
    <type>"Soller"|"..."?</type>
    <length type="NX_FLOAT32" units="cm">{Length of collimator}?</length>
    <soller_angle type="NX_FLOAT32" units="minutes">{Angular divergence of Soller
          collimator}?</soller_angle>
    <horizontal_aperture type="NX_FLOAT32" units="cm">{Horizontal aperture (if
          rectangular)}?</horizontal_aperture>
    <vertical_aperture type="NX_FLOAT32" units="cm">{Vertical aperture (if
          rectangular)}?</vertical_aperture>
    <radius type="NX_FLOAT32" units="cm">{Radius of aperture (if circular)?</radius>
 </NXcollimator>
```

14.1 Issues

14.1.1 Location

Is the distance to the center of the collimator or to the end nearest to the sample? All collimators may not be coincident with the incident beam center, for example, collimators after the sample. For the sake of generality and consistency the “distance” data element should be replaced by a radius (“distance”) and a space angle (theta and phi) in the “standard” coordinate system.

15 NXaperture

```
<NXaperture name="{Name of aperture}">
    <distance type="NX_FLOAT32" units="m">{Distance from sample}</distance>
    <shape>"Rectangular"|"Circular"|"Elliptical"?</shape>
    <horizontal_aperture type="NX_FLOAT32" units="cm">{Horizontal
         aperture}?</horizontal_aperture>
    <vertical_aperture type="NX_FLOAT32" units="cm">{Vertical
         aperture}?</vertical_aperture>
    <radius type="NX_FLOAT32" units="cm">{Radius of aperture (if circular)?</radius>
 </NXaperture>
```

15.1 Issues

15.1.1 Location

All apertures may not be coincident with the incident beam center. For the sake of generality and consistency the “distance” data element should be replaced by a radius (“distance”) and a space angle (theta and phi) in the “standard” coordinate system.

16 NXattenuator

```
<NXattenuator name="{Name of attenuator}">
    <distance type="NX_FLOAT32" units="m">{Distance from sample}</distance>
    <type>{Type of attenuator, e.g. polythene}?</type>
    <thickness type="NX_FLOAT32" units="cm">{Thickness of attenuator along beam
        direction}?</thickness>
    <scattering_cross_section type="NX_FLOAT32" units="barns">{Scattering cross section
        (coherent+incoherent)}?</scattering_cross_section>
    <absorption_cross_section type="NX_FLOAT32" units="barns">{Absorption cross
        section}?</absorption_cross_section>
    <attenuation type="NX_FLOAT32">{Attenuation factor at the nominal beam
        energy}?</attenuation>
 </NXattenuator>
```

16.1 Issues